(2)

ALMOST A FREE LUNCH:
SAS TO STATA DOWNLOAD PROCEDURE

Lynn Ordway, Fred Gurzeler,
and Bill Rogers

October 1989

P-7578

The RAND Corporation, 1700 Main Street, P.O. Box 2138, Santa Monica, CA 90406-2138

PREFACE

This paper documents the process of downloading a SAS dataset to a microcomputer for use with STATA. Inquiries about the STATADEMO diskette that accompanies the documentation should be directed to RAND's Publications Department.

ACKNOWLEDGMENTS

Special thanks to Mary Layne, Ellen Reinisch, Gail Covitt Roberts, Senator John Paul Jones, and Marty Ross (who is partly responsible for STATA).

HOW TO USE THIS MANUAL

Special typefaces and symbols are used throughout this manual to denote the keystrokes you must make on your microcomputer keyboard. These conventions are used in all Computing Information Center (CIC) documentation.

see next page again

Bold — Bold indicates that you must type characters exactly as they are shown.

Italics — Italics indicate that you should type characters like the ones shown. For example, if the instructions read

get *filename*

type the sequence "get" and the name of the file you wish to transfer.

" " — Quotation marks identify filenames, commands, command parameters, and other system prompts or user responses. To avoid confusion, punctuation marks required by the text are placed OUTSIDE the quotation marks. Punctuation marks appearing within the quotation marks are part of the system prompt or user response.

[options] — Braces indicate special options that may, but need not, be added to a command.

d:\> — ">" is the DOS prompt. The letter that precedes the ">" represents the disk drive you are using, e.g., "A:\>" represents the root directory in drive A. In this manual, the prompt "*C:*\>" is listed. Substitute the appropriate letter for the drive you are using (A, B, or C).

% — The percent sign indicates the prompt on the Text Processors and other RAND computers running UNIX.

cue==> — The CUE prompt. A line beginning with "cue==>" indicates that the system is waiting for a CUE command.

<Brackets> — Angle brackets indicate a key with a particular label on it. Thus, if you are told to press

<Ctrl>

press the key labeled "Ctrl". <CR> represents the key marked "Return" or "Enter".

<Key+Key> — Two keys joined by a "+" sign indicate a key combination. The first key listed should be held down while the second key is pressed. Then both are released. For example, if

the instructions show

<Shift+F1>

hold down the <Shift> key, then press the <F1> key, then release both.

<Key> <Key> Two keys in separate brackets indicate a successive key combination. The first key listed should be pressed, released, and then the second key should be pressed and released. For example, if the instructions show

<Esc> <Del>

press the <Esc> key, release it, then press the <Del> key and release it.

CONTENTS

INTRODUCTION

Are you interested in reducing your mainframe charges? How about running statistical analyses during the day or testing your wildest statistical "what ifs" without concern about costs? Well, unless the funding for your project is big enough to make an impact on the national deficit, chances are you're very interested.

Running large jobs on WYLBUR can easily consume large amounts of money that could be better spent elsewhere. Even running a job overnight or on the weekend to save on computing costs can negate its cost effectiveness if time is of the essence. The logical alternative is to do the work on a PC using a program such as STATA.[1] A standard microcomputer has enough memory for a SAS[2] dataset that takes about 20 tracks of WYLBUR data. If you are fortunate enough to belong to a project that can afford to give you a 2mb expanded memory board, your PC can handle a dataset that takes about 100 tracks on WYLBUR.

The problem, however, was that large datasets limited the usefulness of STATA in the RAND environment in the past. But now the new version of STATA can handle datasets with as many as 32,000 observations using a PC with expanded memory and even more on a SUN.[3]

This documentation describes the process of downloading a SAS dataset to your microcomputer for use with STATA. It is also (more or less) a first for a RAND publication; a diskette is included with the basic documentation. The STATADEMO diskette that accompanies this documentation not only contains the public domain AWK-like processor BAWK and several BAWK programs, it also contains an archiving program so you can condense or compact the downloaded datasets, several other programs especially written for this documentation, plus a special (and very useful) bonus program.

The STATADEMO diskette contains the following programs:

- contents - a SAS job that gets contents and means
- jcllines - job control lines that are added to the dictionary
- bawk.exe - an enhanced bawk processor
- contz.b - a program that converts contents into a dictionary
- retsass.b - a program that converts a dictionary into a SAS download program and a STATA dictionary

---

[1]STATA is a trademark of Computing Resource Center.
[2]SAS is a trademark of SAS Institute Inc.
[3]SUN is a trademark of Sun Microsystems, Inc.

- strip.b - a program that strips away prepended sort information and creates the ".dct" and ".sas" files

- res.bat - a batch file that runs the retsass and strip steps

- merge.exe - a program (written by Bill Rogers) used in conjunction with res.bat

- wylpc.exe - a program that, given the DSN filename, the number of lines per observation, and the number of observations, will outline the dataset download procedure

- strip.bat - a batch file that removes the extraneous first and last lines from a downloaded WYLBUR file

- arc.exe - a program that archives the raw datasets once they have been infiled into STATA

The STATADEMO diskette also includes a special bonus program:

- bigsort.bat - sorts files beyond 63k. See Appendix E for details.

This manual presumes knowledge of the RAND computing environment, including experience with WYLBUR, the cue communications package, the esp editor, and a basic knowledge of STATA. If you are not adequately familiar with all of the above, please do not attempt any of these procedures without the presence and supervision of a qualified RANDite.

I. CUSTOMIZING THE STATADEMO DISKETTE

Before you begin, please note the following (very) helpful suggestions:

a) Use at least a Compaq 286 or faster. Following the procedures outlined in this manual can literally take all day if you are working with a very large file on a regular PC.

b) Make sure you have more than enough space on your hard drive. Five megabytes is usually sufficient for even the largest jobs.

The first thing you should do is make a directory called "DOWN" and add it to your path. Next, copy all the files and programs from the STATADEMO diskette to the "DOWN" directory. You may also want to make a directory that you will download your dataset(s) to. You may call it anything you like.

Two files, the one called "contents"

```
//X1043 JOB (4195,60,45),CONTENTS,CLASS=N,MSGLEVEL=(2,0)
// EXEC SAS,OPTIONS='NOCENTER,NONEWS'
//IN DD DISP=SHR,DSN=X.X1043.A4195.URDSN
//SYSIN DD *
PROC CONTENTS DATA=IN._ALL_ NOSOURCE;
PROC MEANS DATA=IN.URSASNAME;
```

and "jcllines"

```
01*//X1043 JOB (4195,60,45),RETRIEVE,CLASS=N,MSGLEVEL=(2,0)
03*//IN DD DISP=SHR,DSN=X.X1043.A4195.URDSN
06*//OUT DD DISP=(NEW,CATLG),UNIT=USER,VOL=SER=TEMP12,
07*//       DCB=(RECFM=FB,LRECL=80,BLKSIZE=3600),
08*//       DSN=X.X1043.A4195.URDSN,
09*//       SPACE=(CYL,(1,1),RLSE)
11*DATA TEMP; SET IN.URSASNAME;
```

need to be modified. Make the following changes to both files:

1. Replace "X1043" with your employee (or user) number. Note that the first "X" in "DSN=X.X1043" should be replaced with the same letter that begins your employee number.

2. Replace "4195" with your account number. Don't forget the "A" in the "DSN=" line.

3. Replace the "45" within the parentheses in line 1 with your bin number.

4. Replace "URDSN" with your dataset name (DSN) and "URSASNAME" with your SAS name.

II. GETTING THE CONTENTS AND MEANS OF THE SAS DATASET

After you've made (and double-checked for accuracy) the changes to "contents" and "jcllines," log on to WYLBUR. You MUST use the following "set" commands whenever you log on to WYLBUR to do any of the steps outlined in this manual:

```
COMMAND ? clear text [or "ct"]
COMMAND ? set uplow
COMMAND ? set slowlist
COMMAND ? set scroll
COMMAND ? set page 23
```

Press <F9> to get the "cue==>" prompt and enter

```
cue==> put contents
```

The file "contents" is now being uploaded to WYLBUR. When the upload has ended, enter

```
COMMAND ? list 1/10
```

This will output the first ten lines of "contents" to the screen. Line one should look similar to this example:

```
//X1043 JOB (4195,60,45),RETRIEVE,CLASS=N,MSGLEVEL=(2,0)
```

After you've uploaded the file to WYLBUR, check for any transmission lossage; sometimes the first 2-5 characters ("//" or even "//X1043") of the first line may not make it from your PC to the computer room. If a lossage occurred, restore the missing characters using WYLBUR's "i" command. After making any necessary repairs, run the job with

```
COMMAND ? run hold
```

Press <F7> to check the job status. The "AWAITING FETCH" status signifies the job's finished executing. You may now "fetch" the job:

```
COMMAND ? fetch [job #] clear
```

Check to see if the job executed without error by entering

```
COMMAND ? list 'CODE'
```

If the "COND CODE" is "0000" and the "COMPLETION CODE:" is "SYSTEM 000" the job was successful (so far). Press <F9> and at the "cue==>" prompt enter

```
cue==> get contents.out
```

Some unnecessary lines need to be edited out of "contents.out" before

continuing. To run esp without logging off WYLBUR,[1] press <F9> and enter the following:

cue==> **edit** [or **run esp**] **contents.out**

Remove the first three or four lines up to (but not including) the line containing "//X1043 JOB (4195,60,45)" etc. and "COMMAND ?" at the end of the file. Exiting from esp will return you to WYLBUR.

---

[1]The procedures in this manual are designed so you do not have to log off WYLBUR until you've downloaded all your datasets.

III. CREATING THE DICTIONARY

The following programs NOT supplied with the STATADEMO diskette must be in your path before doing the procedures in this section:

- find.exe (should be in your "dos" directory)
- sort.exe (should be in your "dos" directory)
- split.exe (should be in your "rbin" directory)

The next step is to create a preliminary dictionary file:

```
COMMAND ? <F9>
cue==> run bawk contz.b contents.out  > urdsn.dd
```

where "urdsn.dd" is replaced with your dataset name (don't forget the ".dd" extension). Next

```
COMMAND ? <F9>
cue==> run esp urdsn.dd
```

and "merge" the file "jcllines" to the beginning of your ".dd" file. There should be one blank line between the end of "jcllines" and the beginning of the dictionary. Next, check the lines in the dictionary that begin with a nine (9) against the other variables;[1] most will be duplicate variables and should be deleted.

The final step is to create a ".dct" and ".sas" file:

```
COMMAND ? <F9>
cue==> run res urdsn
```

where "urdsn" is the ".dd" file. Do not add the ".dd" extension when using RES.BAT; you will get an error if you do. Furthermore, when RES.BAT asks you to "Enter output file name:" be sure to answer "a1," "a2," "a3," etc. and NOT "A1," "A2," "A3," etc. The number of output file names you will be prompted for depends on the size of your file. If the file is small, for example, RES.BAT won't ask beyond "a1."

When RES.BAT has finished executing, enter

```
COMMAND ? <F9>[2]
cue==> run find "NUMBER OF OBSERVATIONS:" contents.out
```

Make note of the number of observations; you will need this information

---

[1]For more information on the SAS dictionary, see Appendix B, "The SAS Dictionary."

[2]By now you've probably realized that control is always returned to WYLBUR after executing a command at the "cue==>" prompt and that you have to press <F9> every time you want to transfer control back to cue.

later. Also get the last card number in the ".sas" file; this is the number of lines per observation which you will also need later.

IV. PREPARING FOR DOWNLOADING

Now that the files "urdsn.sas" and "urdsn.dct" have been created, issue the following commands:

```
COMMAND ? clear text [or "ct"][1]
COMMAND ? <F9>
cue==> put urdsn.sas
```

The file "urdsn.sas" is now being uploaded to WYLBUR. When the upload has ended, enter

```
COMMAND ? list 1/10
```

This will output the first ten lines of "urdsn.sas" to the screen. Line one should look similar to this example:

```
//X1043 JOB (4195,60,45),RETRIEVE,CLASS=N,MSGLEVEL=(2,0)
```

Sometimes part of the first line may not make it through the upload procedure and will have to be restored using WYLBUR's "i" command. Make any necessary corrections and enter

```
COMMAND ? run hold
```

Press <F7> every few seconds to examine the progress of the job. When the job is "AWAITING FETCH" enter

```
COMMAND ? fetch [job #] clear
```

This brings the "run hold" output into the active file, replacing "urdsn.sas" with "urdsn". This is the file that will be downloaded. Continue with

```
COMMAND ? show dsn on temp12 space [dated] [full]
```

The screen output should be similar to

```
TEMP12
 $X.X1043.A4195.URDSN   60 TRACKS (50 USED)
```

If you have "(0 USED)" or "TEMP12" shows up by itself something went

---

[1]This is important; if you do not issue the "clear text" (or "ct") command "urdsn.sas" will be appended to "contents.out" and although everything will appear to run smoothly, the net result will be data garbage. If you are uploading multiple ".sas" files, "ct" before you "put" the file into WYLBUR. If, for some reason, you logged off WYLBUR, don't forget to reenter the "set" commands (see Section II) after logging on again.

wrong. Before you curse this manual, WYLBUR, or computers in general, type

```
COMMAND ? list 'ERR'
```

to check for error codes. You may also want to check your "contents" and "jcllines" files for errors. Sometimes the error may be something as small as a missing comma or semicolon. Refer to the "Error Checklist" in Appendix A for further information.

V. DOWNLOADING TO THE PC[1]

WYLBUR can comfortably download about 9000 lines of raw data at one time. If the dataset is larger than 9000 lines it will have to be downloaded in segments.

The program called WYLPC.EXE (written by Fred Gurzeler), which is included on the STATADEMO diskette, is a useful aid for downloading very large files; it will provide you with a step-by-step breakdown of the download procedure. Please note that WYLPC.EXE will not do the actual downloading; you must ask WYLBUR to do that yourself.

The correct usage for WYLPC.EXE is

```
cue==> run wylpc urdsn nlo no [>lpt1]
```

where "urdsn" is your DSN filename, "nlo" is the number of lines per observation, and "no" is the number of observations. The ">lpt1" option directs the output to the printer instead of the screen so you may have a hardcopy guide to follow.

The following is an example of a small multiple download.

```
cue==> run wylpc urdsn 3 8000

COMMAND ? use urdsn clear
COMMAND ? del 9001/L
COMMAND ? <F9>
cue==> get urdsna.raw

COMMAND ? use urdsn clear skip=9000
COMMAND ? del 9001/L
COMMAND ? <F9>
cue==> get urdsnb.raw

COMMAND ? use urdsn clear skip=18000
COMMAND ? <F9>
cue==> get urdsnc.raw
```

Note: up to five downloads will fit on a PC screen, which is the equivalent of 45,000 total lines. Therefore, you may print the screen after running WYLPC.EXE if you wish. Alsc note that 9000 lines is a maximum; the number of lines downloaded at one time may be less.

Although WYLPC.EXE will provide an easy-to-follow outline of the download procedure, it doesn't hurt to know what is happening during each step. The following is a line-by-line explanation of the above example as it would be used in WYLBUR. When you enter

---

[1]You may also download from WYLBUR to a text processor. If this alternate method excites you more, go to the end of this section.

COMMAND ? **use** urdsn **clear**

you will get one of the following messages:

"urdsn" WON'T FIT, DELTA TOO BIG

or

WORKING FILE TOO BIG. COMMAND TERMINATED

because the dataset is too big (if the dataset is small enough to download all at once you won't get these warnings). Ignore these messages and continue with

COMMAND ? **del** 9001/L

Please note that the number after "del" is always one more than the maximum number of lines per transfer. "L" tells WYLBUR to delete to the "Last" line. Press <F9> after the "COMMAND ?" prompt and enter

cue==> **get** urdsna.**raw**

The "WYLBUR TO PC File Transfer Procedure" is now in effect.

If you are downloading several files, it is important to know in what order the files were downloaded after returning to DOS. A good way to do this is to add a different suffix (a, b, c, etc.) to the filename for each download. Hence, the "a" after "urdsn". Note, too, the ".raw" extension. Since STATA reads its ".dta" files from ".raw" files, all files should be downloaded with the ".raw" extension. WYLPC.EXE will automatically replace any extension your DSN filename may have with the ".raw" extension.

The download may take several minutes. As you watch the data scroll by you may notice a "***possible lossage***" message every now and then. Ignore these for now. After the download has been successfully completed, you will be returned to the "COMMAND ?" prompt. Enter

COMMAND ? **use** urdsn **clear skip** = 9000
[overload error message]
COMMAND ? **del** 9001/L
COMMAND ? <F9>
cue==> **get** urdsnb.**raw**

Because the first 9000 lines have been downloaded, it is not necessary to download them again (obviously). Therefore, the "skip =" command is used for every transfer following the first one. In this example "skip =9000" is used because you want to skip over the 9000 lines that have already been downloaded. The remaining commands are the same as the first download except "urdsn" has a "b" tacked on to it to denote its transfer order. The last transfer is a little different:

```
COMMAND ? use urdsn clear skip = 18000
[no error message]
COMMAND ? <F9>
cue==> get urdsnc.raw
```

First, note that the number skipped has apparently doubled. Actually, the number has increased by 9000 (if there were a fourth download "skip" would equal 27000, a fifth "skip" would equal 36000, etc.). Furthermore, since this is the last transfer, the remaining number of lines is small enough to avoid the error message. And finally, it is not necessary to delete lines for the last transfer and so the delete command is omitted. Note, too, that a "c" has been added to "urdsn" to distinguish it from the previously downloaded datasets.

With the exception of the last transfer, all of the downloaded files should be the same size. If they are not, there was probably a transmission failure and the file may have to be downloaded a second time. To check the files that have been downloaded to the PC without exiting WYLBUR, enter the following commands:

```
COMMAND ? <F9>
cue==> dir *.raw
```

If the file sizes are all equal, the downloading has proceeded smoothly. If you have to download a file again, simply download that segment again; you do not have to start over. If, for example, the second download wasn't successful, start with "use urdsn clear skip=9000" at the "COMMAND ?" prompt.

After the last transfer you are ready to log off:

```
COMMAND ? logoff clear
COMMAND ? <F9>
cue==> exit
C:\>
```

DOWNLOADING FROM WYLBUR TO A TEXT PROCESSOR

To download to a tp, do everything up to and including the "run hold" command in Section IV. Do not "fetch" the job(s). Type

```
COMMAND ? use urdsn.raw clear
COMMAND ? save urdsn.raw replace
COMMAND ? show dsn space on temp12 full
```

The "LRECL" should equal 3665. You may "use" and "replace" more than one job if you like. Log off WYLBUR and log on to your tp. At the prompt enter

```
tp5% retrieve -v -dsn=X.X1043.A4195.urdsn.raw urdsn.raw
```

replacing the user ("X.X1043") and account number ("A4195") with the correct ones.

The "retrieve" command works in the background (which is a good thing because it is very slow) and so once you've okayed a retrieve, you may retrieve another file, do something else, or even log off. Typing "print -q" will show the spooled IBM requests and "du -a [path]urdsn.raw" is one way to check on the incredible growing retrieved file.

For more information on downloading from WYLBUR to the tp, type "man retrieve" at the tp prompt. If you are pursuing this method of downloading, the remainder of this documentation really doesn't apply anymore, although you may want to finish reading it anyway.

VI. WORKING WITH THE DATASET ON THE PC

The downloaded dataset will have two extra lines ("1 unn page=0" and "COMMAND ?") at the beginning and the end which need to be removed before the dataset can be read into STATA. The program included on the STATADEMO diskette called STRIP.BAT (written by Mary Layne) removes these lines. Run STRIP.BAT as follows:

```
C:\>strip urdsn[a].raw
```

where "[a]" is the suffix indicating the file is one segment of a larger file (a one-shot downloaded file shouldn't have a suffix). You should get an output similar to the following on your screen:

```
C:\>FIND/V "1 unn page=0" < urdsn[a].raw | FIND/V "COMMAND ?" >
urdsn[a].raw
```

"Strip" the rest of the files (suffix "b," "c," etc.) if you had more than one download. You may also want to type or edit the files to make sure you had a good transmission.

You are now ready to read this dataset into STATA. You don't have to read it in all at once. You can cut down the number of variables or read in the observations a few at a time by entering in STATA:

```
. set prefix c:\directory
. infile using c:\directory\urdsn[?].raw[, auto]
```

If you decide to cut down the number of variables, you can do this by eliminating the lines that have these variables. Don't eliminate the _newline directives, though, or STATA will read values from the wrong records. This is a common error.

If you decide to read it in chunks, do all of the "infile" statements in the same STATA session. The reason for this is that you want the same values assigned to the data labels.

Be sure to SAVE your dataset as a STATA ".dta" file. The "auto" option, by the way, causes STATA to create value labels from the non-numeric data it reads. Refer to the STATA manual for further information.

VII. ARCHIVING THE DATASET

Since a very large dataset takes up valuable disk space, it is always a good idea to archive the files so you can delete the ".raw" files and free up some bytes. This is done using the program ARC.EXE included on the STATADEMO diskette. After you've constructed a dataset you feel you can live with, archive your files by entering

```
C:\>arc a urdsna urdsna.raw
```

Note that the first filename is entered without an extension. This is because ARC.EXE will add the extension ".arc" to the name. The second filename is the file you want archived. ARC.EXE will then output (in stages) to the screen

```
Creating new archive: URDSNA.ARC
Adding file:   URDSNA.RAW    analyzing, crunching, done.
```

You can also archive all the files into one file by typing

```
C:\>arc a urdsn urdsn?.raw
```

ARC.EXE will compress the "urdsn?.raw" files, in order, into a file named "urdsn". Note: although archiving can compress a file up to 25% of its original size, it does not touch the original file. Therefore, make sure you have enough disk space to accommodate the ".arc" files. Also, redirecting the archived output (to drive A:, for example) is not a very stable process, i.e., it may or may not work. Instead, archive to the C: drive and then copy the "*.arc" file to A: drive.

To restore the archived file to its original state, type

```
C:\>arc x [or e] urdsn
Extracting file: urdsna.raw
Extracting file: urdsnb.raw
etc.
```

VIII. HELPFUL SUGGESTIONS FROM THE AUTHORS

Bill Rogers:

In general, it's a good idea to put each project on its own directory or floppy. I put the STATDEMO programs in a directory called "down" on all of my machines, and the batch files go into "bin" (or "rbin"). This has to be in the path.

Also, it is not a good idea to put the utilities on the diskette with the data. For one thing, you use up a fair amount of space on the diskette, and you also clutter it up with files.

Lynn Ordway:

Unlike a research environment in which all use SAS and save online and possibly hard copies of output, with STATA there is a serious danger when analysts leave no paper trail. In our office, we have struggled to maintain records of "how we got those numbers" by keeping logs of all analysis in a separate "*.log" directory.

I also keep a directory with all my ".do" files so that I can modify and reuse them.

Fred Gurzeler:

Keep your dataset names consistent across systems by using the DOS filename convention, i.e., eight characters plus a three character extension. I highly recommend that your dataset names have no more than seven characters; the eighth may be necessary for the extra letter used for segmented downloads of a large file.

Appendix A

ERROR CHECKLIST

If the job does not run correctly, check to see if you've satisfied the conditions below. Some errors are not self-evident; a job will seem to be executing just fine with item number two, for example, until you fetch the output and discover it didn't execute at all.

1. Make sure you've correctly modified the "contents" and "jcllines" files.

2. Does the DSN already exist? If so, you will have to "scratch" the data set. Avoid using duplicate names on the direct access volume. Always "scratch" a job if you plan on running it more than once. It also doesn't hurt to "purge" an unsuccessful job, either.

3. Is the preliminary SAS dictionary complete? If the SAS variable name does not have a data type and means or frequency indentifier (see Appendix B for more information) the job won't execute correctly. A missing variable name will cause a job to fail, too.

4. Does the job have enough time to run? Sixty seconds is usually sufficient, but large jobs may need more time.

Appendix B

THE SAS DICTIONARY

A typical dictionary entry looks like this:

```
3        AGE          i m      AGE OF PATIENT
```

What does each component mean? The first number ("3") is the field width and represents the number of character columns that will be used to store the variable during transmission. If the number is too small, you will lose precision in the download. If it's too large, the dataset will be too unwieldly to transmit or save.

"AGE" is the SAS variable name which will be translated to lowercase letters and become the STATA variable name.

The "i" is the data type and may be integer, numeric, or character. "Integer" means integer data in the range -3200 to +32000. STATA stores this in 2 bytes. "Numeric" refers to data with fractional parts or data outside the range (-32000,+32000). "Character" is any character data.

The "m" is a means or frequencies identifier and is used with software that prepares automatic tabulations. It is not important here.

The last item is the variable label. Keep in mind that while SAS allows up to 40 characters, STATA will truncate everything beyond 30 characters.

There are two replications of the dictionary in the ".dd" file. One is created by analyzing the PROC CONTENTS and one is created by analyzing the PROC MEANS. The CONTENTS versions tend to have lengths of 9 for numeric data, which will conserve accuracy but take up more space than necessary.

A potential problem exists if the data include variables that have a MIN equal to an integer and a MAX equal to another integer when the actual data takes on values BETWEEN the two integers.

Appendix C

FIELD WIDTHS

The program that analyzes PROC MEANS output can be too aggressive about making the field width small. It does this because it looks only at the MIN and the MAX in PROC MEANS. Sometimes a variable has a MIN like 0 and a MAX like 1, so it looks like a dummy variable when it actually takes values in between, like 0.345657.

For example, this problem came up on the DRG study; variables which had a MIN=0 and a MAX=1 took on values in between 0 and 0.5. Since we had left too little space in the type column, the values between 0 and .5 were converted to 0 and the values at .5 were converted to 1.

You should review any field widths you use from PROC MEANS to see if this is a problem. Of course, you may not know your data well enough. If not, SAS will print a "*" in the offending column. STATA will try to read the "*" during the infile step and print an error message. If you don't know the data, however, the best solution is to use the more conservative lengths from PROC CONTENTS. (These procedures are continually being updated and user suggestions are welcome.)

To double-check that this has not occurred, after STEP FIVE compare the means given by the SAS PROC MEANS with those given by STATA'S "summary" command.

Appendix D

CHARACTER DATA

You may also want to consider what you want to do with character data. The default choice of the download software is to treat each distinct character value as a category and define STATA value labels to hold the string equivalent. This works with most of the STATA commands.

A second choice is treat the character value as a STATA string (version 1.64 and higher). If you want this, change the type of the variable in the STATA dictionary from int to strnn where nn is the maximum length of the string, and delete the value label designator (:varname).

The third choice is to delete or otherwise modify the entry. In the MOS, we have a 7 character identifier of which the first 6 characters are numeric and the last is a check digit, which is often a letter. So we change the entry for patid to numeric by deleting the "int" type and changing the format from %7s to %6g.

For example, this:

```
_column(1) int patid   :patid    %7s "Patient ID"
```

will become this:

```
_column(1)  patid    %6g "Patient ID"
```

Appendix E

BIGSORT.BAT

BIGSORT.BAT, although not a part of this documentation, is included with the STATADEMO diskette because we feel you will find it quite useful. A by-product of RES.BAT, BIGSORT.BAT (as the name implies) can sort files beyond 63k, which is the limit of the DOS sort. Like its cousin RES.BAT, BIGSORT.BAT requires MERGE.EXE, SORT.EXE, and SPLIT.EXE to be in your path. The correct usage is

C:\> **bigsort** *filename*[*.ext*] *filename*

Do not use an extension for the second filename; BIGSORT.BAT will add the extension ".srt" to distinguish the new, sorted file from the original, unsorted file. Furthermore, when BIGSORT.BAT asks you to "Enter output file name:" be sure to answer "a1," "a2," "a3," etc. and NOT "A1," "A2," "A3," and so forth. The number of output file names you will be prompted for depends on the size of your file.

Just how big a file can BIGSORT.BAT sort? We're not sure. A 320k file was sorted with room to spare, although 500k is probably the outer limit.

BIGSORT.BAT was written by Fred Gurzeler and Bill Rogers.